Flappy Bird és un joc de mòbil que es va popularitzar l'any 2013 que consisteix en controlar un ocell que ha de volar entre una sèrie de tubs que apareixen a la pantalla. Per mantenir-lo en vol, el jugador ha de tocar la pantalla enlairant l'ocell lleugerament, si no es toca, l'ocell cau a causa de la gravetat. L'objectiu principal és passar entre els tubs sense xocar-hi ni caure al terra i cada cop que l'ocell passa amb èxit entre dos tubs, el jugador guanya un punt.

--------------------------------------------------------------------------------------------------------------------------

Les diferents combinacions de inputs, mutacions i poblacions generen diferents comportaments en l’evolució i l’aprenentatge de la xarxa neuronal. A continuació s’explica breument com es comporta la xarxa neuronal depenent de les variables que li introduïm.

Imatge que conté text, escriptura a mà, Font, línia

Descripció generada automàticamentQuan la xarxa neuronal només coneix la posició Y de l'ocell (input 0), els resultats són completament aleatoris anant desde 0 fins a 4000 de fitness. Mai s'apropa a una dada superior 4000 de fitness, ja que no té la informació necessària per arribar al seu objectiu, degut a que amb les dades que posseeix no pot saber en quina posició es troba el forat de l'obstacle i l’unic que pot fer es evitar caure al terra i estar a una altura en la qual pot haver un forat. (fiacar al peu de foto el fitness es petit i comparat amb altres gràfiques que no ha arribat gents lluny)

Imatge que conté text, captura de pantalla, línia, Paral·lel

Descripció generada automàticamentD’altre banda, quan només li donem la posició Y de l’obstacle (Input 1), la xarxa neuronal nomes sap a quina alçada està el forat de l'obstacle i no la posició de l'ocell, per la qual cosa no és capaç de superar-lo ni de manera aleatòria, com sí que passava amb l'input 0, perquè esta jugant completament a cegues.

Quan la xarxa neuronal coneix la posició Y de l'ocell i la posició Y de l'obstacle (input 0-1), l'aprenentatge de la xarxa és més ràpid i efectiu, ja que té la informació necessària per completar la tasca. D’altre banda, un sol input que relaciona aquestes dos variables de forma directa, és més eficient, millora la velocitat d’aprenentatge i redueix la complexitat de la xarxa neuronal oferint uns millors resultats acord amb les gràfiques. (sha de fer gràfiques)

Quan oferim la posició Y de l'ocell i la posició Y i X de l'obstacle (Input 0-1-2) el temps d'aprenentatge augmenta respecta amb la configuració d'inputs 0-1 degut a l’augment de complexitat de la xarxa neuronal

Si combinem la posició Y de l'ocell, la posició Y de l'obstacle i la velocitat Y de l'ocell (Input 0-1-3) passa a ser una configuració menys efectiva que la anterior, ja que l'aprenentatge es ralentitza més per la presència de la velocitat Y, un input que la xarxa tendeix a ignorar o eliminar.

Imatge que conté text, línia, Trama, captura de pantalla

Descripció generada automàticamentQuan ajuntem ja tots els inputs posició Y de l'ocell, posició Y i X de l'obstacle, velocitat Y de l'ocell (input 0-1-2-3) alenteix encara més l’aprenentatge i passa a ser una de les xarxes neuronals menys optimes, traient de banda les que no tenen prou informació, degut a que ha de ignorar el excés d’informació.

Es a dir, com més informació innecessària rep la xarxa neuronal, i més complexa és aquesta, l’aprenentatge s’alenteix i/o limita degut a que la xarxa neuronal és lineal i esta pensada a resoldre problemes lineals no massa complexos com el flappy bird.

Mutació no estructural (Mutació 0):

Mutació estructural (Mutació 1):

Mutació crossover (Mutació 2):

Imatge que conté text, línia, captura de pantalla, Trama

Descripció generada automàticamentLa aplicació de la mutació no estructural (Mutació 0) en general provoca unes puntuacions de fitness més altes i un aprenentatge més ràpid quan la apliquem sola, però quan es tracta de generacions on es dona un excés d’informació o on hi falta informació l’aprenentatge de la mutació s’alenteix degut a que no pot eliminar ni crear connexions i el ha de perdre el temps aproximant el valor d’una connexió a 0. D’altre banda quan la ajuntem amb més mutacions perd eficiència.

Imatge que conté text, línia, Font, captura de pantalla

Descripció generada automàticamentQuan apliquem la mutació estructural (Mutació 1) , en si sola només és genera un resultat per sobre de la majoria quan es tracta de aplicar-les en generacions amb excés o falta de inputs, però en aquestes mateixes també és superada per altres mutacions passat unes quantes generacions, degut a que inicia amb un primer salt però després s’estanca. D’altre banda quan l’ajuntem amb més mutacions, perd eficiència en generacions amb inputs no gaire eficients, però en guanya quan es donen bons inputs d’entrada assegurant que superi a quan esta la mutació sola.

D’altre banda, la implicació de la mutació crossover (Mutació 2) no provoca canvis notables en la majoria de partides però a vegades pot ser molt bona o molt dolenta sense cap patró aparent.

En quant a la població, el que fa es augmentar la velocitat d’aprenentatge d’una generació en funció de la població que hi hagi, a causa de la presència de més xarxes neuronals que permeten que aprengui més de pressa.

Cal recalcar que no tots els gràfics ens han donat aquestes conclusions al peu de la lletra degut a la quantitat de variables que poden afectar a com es comporta cada generació, però l’anàlisi dels gràfics fets anteriorment busca ser el més acurat possible.

------------------------------------------------------------------------------------------------------------------------------

Línies a futur

Mirant cap al futur, la recerca realitzada en aquest treball obre la porta a noves línies d'investigació que podrien ampliar aspectes en els que no s’ha pogut aprofundir o donar-los suficient rellevància, i oferir una continuïtat i ampliació a la recerca fins ara feta.

El primer aspecte a ampliar consistiria en passar d’una xarxa neuronal lineal, plantejada en un inici per resoldre problemes simples, a una xarxa neuronal complexa, i estudiar les diferències entre els dos tipus de xarxes i els resultats obtinguts. Al utilitzar una xarxa neuronal complexa també es podrien plantejar nous tipus de paràmetres i variables a analitzar, la qual cosa faria que els jocs que podrien ser analitzats, a la vegada també podrien ser més complexos. En l’aspecte de l’aprenentatge, també es proposa poder aplicar més d’una mutació per IA i calibrar el nombre d’aquestes per veure si millora algun paràmetre d’aquest aprenentatge.

Per exemple, per fer el problema més complex es podria implementar en el joc una funció en la que aparegués una moneda en una alçada aleatòria i que l’ocell hagués de veure si es viable agafar la moneda o si en agafar-la es moriria i en funció d’això decidir que fer. En aquest sentit s’hauria d’adaptar la xarxa neuronal per determinar la forma correcta d’administrar el fitness i també determinar la millor forma d’entrenar la IA, si per etapes separades per cada dificultat concreta o si directament amb totes les dificultats alhora.

Augmentar les dificultat i sobretot passar d’una xarxa lineal a una complexa implica poder disposar de molta més capacitat de processament de dades (que els equips informàtics dels que disposem fins ara no ens poden proporcionar) per l’obtenció dels resultats. Per agilitar aquest procés s’hauria de millorar la forma de la recollida de dades, ordenant i visualitzant-les d’una forma més senzilla i recopilant-les de forma més rapida i eficient. Per agilitar aquest procés es proposen dos principals camins a seguir:

* Disposar d’equips informàtics més potents i amb molta més capacitat de processament matemàtic (GPU).
* Passar d’un processament i recopilació de les dades de resultats centralitzat a un processament i recollida de dades descentralitzada o distribuïda.

La primera opció és la més immediata i no requeriria pràcticament la modificació dels processos actuals més enllà de les optimitzacions que es creguessin convenients. La principal dificultat d’aquesta opció és el seu cost econòmic i la poca escalabilitat que ofereix a la llarga.

La segona opció, tot i que requereix més esforços de desenvolupament, permetria a un baix cost, obtenir molta més capacitat de processament al repartir de forma distribuïda en diferents sistemes de baixa capacitat la tasca d’obtenció dels resultats.

Per dura a terme aquesta segona opció, s’hauria de repartir, entre quants més col·laboradors voluntaris es pogués, una còpia del joc ja parametritzat amb una paràmetres definits. Els resultats obtinguts es podrien transferir en format JSON (ja implementat al sistema actual) a un sistema central que s’encarregaria de recollir les dades per processar-les, visualitzar-les i analitzar-les en conjunt. Aquest mecanisme es podria fer cíclic per tal d’anar realimentant els jocs amb nous paràmetres a partir dels resultats analitzats.

Tot i afegir més complexitat i requerir més esforç aquesta segona opció permetria a la llarga obtenir més capacitat de processament. Si el projecte busques no només l’anàlisi de resultats sinó també l’aprenentatge de la IA de forma continua, potser la complexitat requerida la podria fer inviable i s’hauria d’optar per la primera opció o una combinació de les dues.

Una darrera línia a futur podria consistir en adaptar la IA del joc en forma de mòdul de tal manera que es pogués utilitzar en altres programes de recerca o directament en jocs dels que es volgués dotar d’una IA per exercí funcions de personatges no jugadors. Per assolir aquest objectiu a banda d’extreure la IA del joc i donar-li forma de mòdul, s’hauria d’establir un estàndard per tal de que el mòdul es pogués acoblar en els sistemes de destí. L’adaptació universal seria molt complexa, però creiem que l’adaptació a plataformes com Godot no hauria de ser excessivament difícil.